Data-Driven Decision-Making In Enterprise Applications

Linear Programming

Rainer Schlosser

Hasso Plattner Institute (EPIC)

April 30, 2020

Decision-Making Using Linear Programming

Given Decision Problem → Formulated as LP Model → Standard LP Solver → Obtain Optimal Solution

Decision-Making Using Linear Programming

Given Decision Problem	→	Formulated as LP Model	→	Standard LP Solver	→	Obtain Optimal Solution
Tough		Magic		Easy		Awesome!

Linear Programming & Focus

- **What is a Linear Program?**
- Theoretical Foundations
- Standard Solution Algorithms
- **Tricks to formulate decision problems as LP**
- **Examples, Examples, Examples, . . .**

What is a Linear Program?

Decision variables $x_1, x_2, \ldots$	The controls to be determined.
Constraints via $\leq, =, \geq$	Expressed linearly in $x_1, x_2, \ldots$.
One objective: max or min	Expressed linearly in $x_1, x_2, \ldots$.

What is a Linear Program?

Decision variables $x_1, x_2, \ldots$	The controls to be determined.
Constraints via $\leq, =, \geq$	Expressed linearly in $x_1, x_2, \ldots$.
One objective: max or min	Expressed linearly in $x_1, x_2, \ldots$.

- Does the solver help you to define the right LP?
- How to formulate a given problem as an LP?
- What does "expressed linearly" means?

Which Terms are Linear in the Variables $x_1, x_2, \ldots$?

$$3x_1 - 2x_2, \quad x_1^{-3}, \quad a \cdot \ln(x_1), \quad \ln(a) \cdot x_1$$

$$a \cdot x_1, \quad a \cdot b \cdot x_1, \quad a \cdot x_1^{2}, \quad \sqrt{\ln(a)} \cdot \frac{x_1 + x_2}{b} \cdot \sin(a^2)$$

$$|x_1|, \quad \max(x_1, 5), \quad x_1^{2} / x_1, \quad (x_1 - 3) \cdot (x_2 + 3), \quad a^2 \cdot x_1,$$

$$x_1 \cdot x_2, \quad x_1 / x_2, \quad 1_{\{x_1 = 5\}} := \text{ if } x_1 = 5 \text{ then } 1 \text{ else } 0$$

Which Terms are Linear in the Variables $x_1, x_2, \ldots$?

$3x_1 - 2x_2$, x_1^{-3}, $a \cdot \ln(x_1)$, $\ln(a) \cdot x_1$

$a \cdot x_1$, $a \cdot b \cdot x_1$, $a \cdot x_1^2$, $\sqrt{\ln(a)} \cdot \frac{x_1 + x_2}{b} \cdot \sin(a^2)$

$|x_1|$, $\max(x_1, 5)$, x_1^2 / x_1, $(x_1 - 3) \cdot (x_2 + 3)$, $a^2 \cdot x_1$,

$x_1 \cdot x_2$, x_1 / x_2, $1_{\{x_1 = 5\}} :=$ if $x_1 = 5$ then 1 else 0

Example of a Linear Program

Objective: $$\max_{x_1, x_2 \in \mathbb{R}} 2 \cdot x_1 + 3 \cdot x_2$$

Constraints: $$0.5 \cdot x_1 \leq 4 - x_2 ,$$

$$0 \leq x_1 \leq 3 , \quad x_2 \geq 0$$

Example of a Linear Program

Objective: $$\max_{x_1, x_2 \in \mathbb{R}} 2 \cdot x_1 + 3 \cdot x_2$$

Constraints: $$0.5 \cdot x_1 \leq 4 - x_2,$$

$$0 \leq x_1 \leq 3, \quad x_2 \geq 0$$

- Only use linear expressions
- Only use $\leq$, $=$, $\geq$ in the constraints ($<$, $\neq$, $>$ are not allowed!)
- Is such an LP understandable for all solvers?

Example of a Linear Program in Standard Form

Objective: $\max_{x_1, x_2 \geq 0} 2 \cdot x_1 + 3 \cdot x_2$

subject to $0.5 \cdot x_1 + 1 \cdot x_2 \leq 4$

$$1 \cdot x_1 + 0 \cdot x_2 \leq 3$$

Standard form:

- Linear combinations of (*non-negative*) variables
- Use $\leq$ (or $=$) in all constraints
- No variables on the "*right-hand side*" of the constraints

Example of a Linear Program in Standard Form

Objective: $$\max_{x_1, x_2 \geq 0} 2 \cdot x_1 + 3 \cdot x_2$$

s.t. $$0.5 \cdot x_1 + 1 \cdot x_2 \leq 4$$

$$1 \cdot x_1 + 0 \cdot x_2 \leq 3$$

$$\max_{x_1, x_2 \geq 0} \vec{c}\,'\vec{x} \quad \text{s.t.} \quad A \cdot \vec{x} \leq \vec{b}$$

with $\vec{c} := (2,3)$, $\vec{b} := (4,3)$,

$$A := \begin{pmatrix} 0.5 & 1 \\ 1 & 0 \end{pmatrix}$$

Example of a Linear Program in Standard Form

Objective: $$\max_{x_1, x_2 \geq 0} 2 \cdot x_1 + 3 \cdot x_2$$

s.t. $$0.5 \cdot x_1 + 1 \cdot x_2 \leq 4$$

$$1 \cdot x_1 + 0 \cdot x_2 \leq 3$$

$$\max_{x_1, x_2 \geq 0} \vec{c}\,'\vec{x} \quad \text{s.t.} \quad A \cdot \vec{x} \leq \vec{b}$$

with $\vec{c} := (2,3)$, $\vec{b} := (4,3)$,

$$A := \begin{pmatrix} 0.5 & 1 \\ 1 & 0 \end{pmatrix}$$

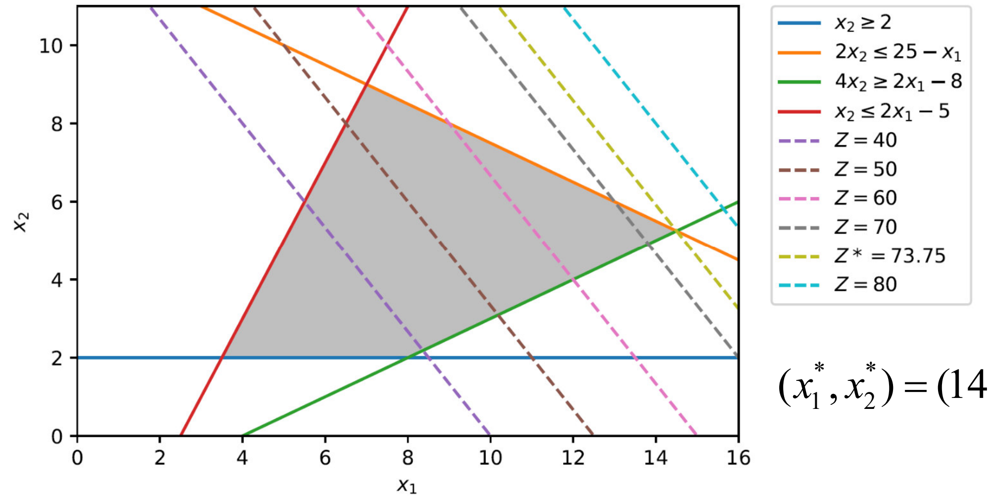
Optimal Solution: Computed by solvers using standard algorithms

For further reading see: Simplex Algorithm,

Convex Polyeder Theory, etc.

Graphical Solutions (Continuous LP)

Example: $\max_{x_1 \geq 0, x_2 \geq 2} 4 \cdot x_1 + 3 \cdot x_2$ s.t. $2 \cdot x_2 \leq 25 - x_1$, $4 \cdot x_2 \geq 2x_1 - 8$, $x_2 \leq 2x_1 - 5$

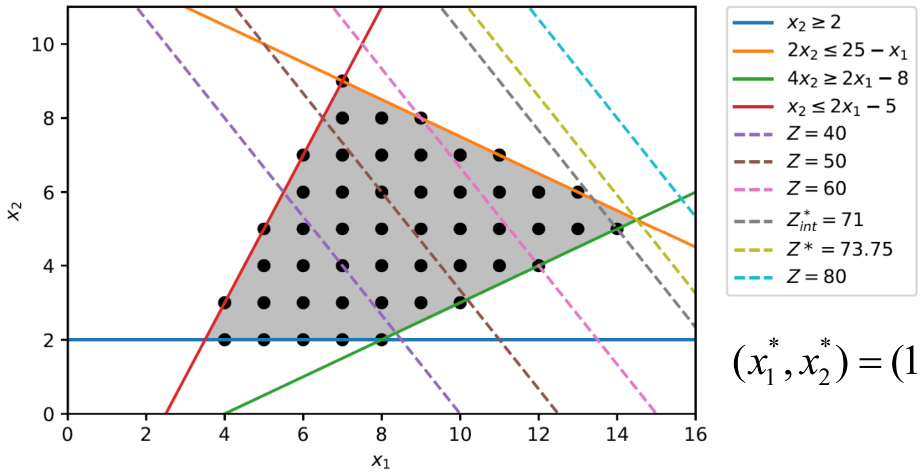


$(x_1^*, x_2^*) = (14.5, 5.25)$

- **Continuous LP:** One corner of the feasible space is always optimal!

Graphical Solutions (Integer LP)

Example: $\max\limits_{\substack{x_1=0,1,2,\ldots \\ x_2=2,3,4,\ldots}} 4\cdot x_1 + 3\cdot x_2$ s.t. $2\cdot x_2 \le 25 - x_1,\ 4\cdot x_2 \ge 2x_1 - 8,\ x_2 \le 2x_1 - 5$



$(x_1^*, x_2^*) = (14, 5)$

- **Integer LP**: Use discrete variables $x_1, x_2, \ldots \in \{0,1\}$ *or* $\in \{0,1,2,\ldots\}$

  Optimal solutions via **integer** solvers (see "Branch & Bound")

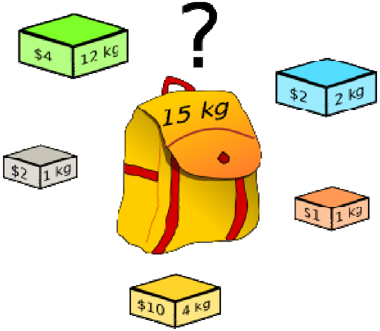
Examples, Examples, Examples

- I Knapsack Problem
- II Magic Square
- III Matrix Inversion
- IV Project Assignment Problems
- V Mixed Equilibria (Game Theory)

I Knapsack Problem

Given parameters:

N	Number of potential items
u_i	Utility of packing item i, $i=1,\ldots,N$
s_i	Space of item i, $i=1,\ldots,N$
C	Capacity of the knapsack



Problem: Decide which items to take to maximize total utility, while not exceeding the knapsack's capacity

I LP Model for the Knapsack Problem

Variables: x_i binary variable whether to pack item i, i=1,…,N

LP: $$\max_{x_1,\dots,x_N \in \{0,1\}} \sum_{i=1,\dots,N} u_i \cdot x_i$$ maximize total utility

I LP Model for the Knapsack Problem

Variables: x_i binary variable whether to pack item i, i=1,…,N

LP: $$\max_{x_1,\dots,x_N \in \{0,1\}} \sum_{i=1,\dots,N} u_i \cdot x_i$$ maximize total utility

s.t. $$\sum_{i=1,\dots,N} s_i \cdot x_i \le C$$ satisfy budget constraint

Let the solver do the rest!

Done.

I Implementation of the Knapsack LP (AMPL)

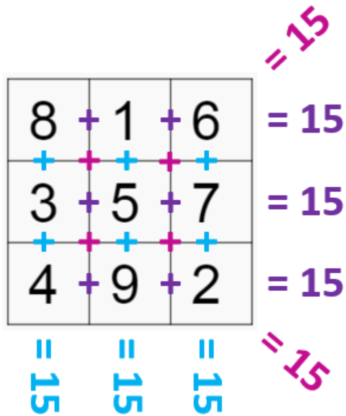
$$\max_{x_1,\ldots,x_N \in \{0,1\}} \sum_{i=1,\ldots,N} u_i \cdot x_i \qquad \text{s.t.} \qquad \sum_{i=1,\ldots,N} s_i \cdot x_i \le C$$

```
reset;
param C := Uniform(50,100);                     # capacity
param N := Uniform(100,200);                    # number of items
param u {i in 1..N} := Uniform(3,8)             # utility of item i
param s {i in 1..N} := Uniform(1,2)             # space of item i

var x {i in 1..N} binary;                       # binary variables

maximize   LP: sum{i in 1..N} u[i]*x[i]; # objective

subject to NB: sum{i in 1..N} s[i]*x[i] <= C;  # budget con.

solve; display x;                               # obtain solution
```

II Magic Square

$N = n^2$ Numbers i=1,...,N to be filled in a square of size $n \times n$, $n \geq 3$

Problem: Assign the numbers i=1,...,N to the cells of a square such that the sum of each row and each column is the same.

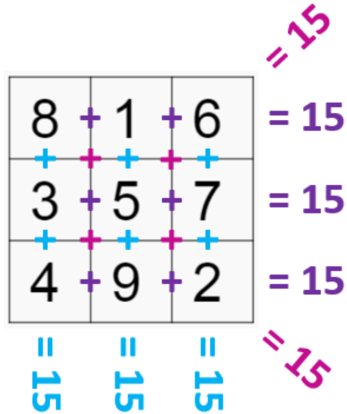


Task: For n=3 the sum is 15. What is the magic number for n=4?

II Magic Square

$N = n^2$ Numbers i=1,...,N to be filled in a square of size $n \times n$, $n \geq 3$

Problem: Assign the numbers i=1,...,N
to the cells of a square
such that the sum of each row
and each column is the same.



Note: The magic number is $C := N \cdot (N+1)/2/n = n \cdot (n^2+1)/2 \overset{n=3}{=} 15$, $\overset{n=4}{=} 34$

II LP Model for the Magic Square

Variables: $x_{i,j,k}$ binary variable whether number k, k=1,...,N is assigned to row i and column j, i,j=1,...,n

II LP Model for the Magic Square

Variables: $x_{i,j,k}$ binary variable whether number k, k=1,...,N is assigned to row i and column j, i,j=1,...,n

LP: $$\max_{x_{i,j,k} \in \{0,1\}^{N \times n \times n}} \sum_{i=1,\dots,n,\ j=1,\dots,n,\ k=1,\dots,N} x_{i,j,k}$$ (no objective is needed!)

s.t. $$\sum_{i=1,\dots,n,\ k=1,\dots,N} k \cdot x_{i,j,k} = C$$ for all j=1,...,n

$$\sum_{j=1,\dots,n,\ k=1,\dots,N} k \cdot x_{i,j,k} = C$$ for all i=1,...,n

II LP Model for the Magic Square

Variables: $x_{i,j,k}$ binary variable whether number k, k=1,…,N is assigned to row i and column j, i,j=1,…,n

LP: $$\max_{x_{i,j,k}\in\{0,1\}^{N\times n\times n}} \sum_{i=1,\dots,n,\; j=1,\dots,n,\; k=1,\dots,N} x_{i,j,k}$$ (no objective is needed!)

s.t. $$\sum_{i=1,\dots,n,\; k=1,\dots,N} k\cdot x_{i,j,k} = C$$ for all j=1,…,n (column sums)

$$\sum_{j=1,\dots,n,\; k=1,\dots,N} k\cdot x_{i,j,k} = C$$ for all i=1,…,n (row sums)

$$\sum_{k=1,\dots,N} x_{i,j,k} = 1$$ for all i,j=1,…,n (unique cell)

$$\sum_{i=1,\dots,n,\; j=1,\dots,n} x_{i,j,k} = 1$$ for all k=1,…,N (each k once)

Hints

- At first: **Think**! Do not start to implement too early
- Do not hope that the LP/solver thinks in your favour!
- Remember: You have to **force** the solver to do the right thing
- Try as hard as you can to formulate the problem **linearly**
- Do not hesitate to use many variables and constraints
- 95% of the work is setting up the LP, 5% is implementation!

III Matrix Inversion

$a_{i,j}$ Coefficients of a given matrix A of size $n \times n$, $i,j=1,\ldots,n$

Problem: Find the **inverse matrix** A^{-1} of matrix A

determined by its coefficients $x_{i,j}$, $i,j=1,\ldots,n$.

Note: The inverse A^{-1} is uniquely determined by satisfying:

$$A^{-1} \times A = A \times A^{-1} = I := \begin{pmatrix} 1 & & 0 \\ & \ldots & \\ 0 & & 1 \end{pmatrix}$$

III LP Model for Matrix Inversions

Variables: $x_{i,j}$ continuous variable for the coefficient of row i and column j of the inverse matrix A^{-1}, $i,j=1,\ldots,n$

LP:

III LP Model for Matrix Inversions

Variables: $x_{i,j}$ continuous variable for the coefficient of row i and column j of the inverse matrix A^{-1}, $i,j=1,\ldots,n$

LP: $\max\limits_{x_{i,j}\in\mathbb{R}^{n\times n}} \sum\limits_{i,j=1,\ldots,n} x_{i,j}$ (objective is optional!)

s.t. $\sum\limits_{k=1,\ldots,n} a_{i,k}\cdot x_{k,j} = 1_{\{i=j\}}$ for all $i,j=1,\ldots,n$ $\left(A\cdot X = I\right)$

III Implementation of the Matrix Inversion LP

Variables: $x_{i,j}$ continuous variable for the coefficient of row i and column j of the inverse matrix A^{-1}, $i,j=1,\ldots,n$

LP: $$\max_{x_{i,j}\in\mathbb{R}^{n\times n}} \sum_{i,j=1,\ldots,n} x_{i,j}$$ (objective is optional!)

s.t. $$\sum_{k=1,\ldots,n} a_{i,k}\cdot x_{k,j} = 1_{\{i=j\}}$$ for all $i,j=1,\ldots,n$ $\quad (A\cdot X = I)$

```
param n := 5;                                       # size of A
param a {i in 1..n,j in 1..n} := Uniform(-1,1);    # coeff. of A
var   x {i in 1..n,j in 1..n};                      # coeff. of A^-1

subject to NB{i in 1..n,j in 1..n}:                 # identity
      sum{k in 1..n} a[i,k]*x[k,j] = if i=j then 1 else 0;

solve; display x;                                   # solution output
```

IV Project Assignment Problem

Given parameters:

N	Number of workers/projects
$w_{i,j}$	Willingness of worker i=1,...,N to take project j=1,...,N

Problem: Decide how to distribute all projects in order to maximize total welfare, while assigning one project to each worker

IV LP Model for the Project Assignment Problem

Variables: $x_{i,j}$ binary variable whether project i, i=1,...,N

is assigned to worker j, j=1,...,N

LP:

IV LP Model for the Project Assignment Problem

Variables: $x_{i,j}$ binary variable whether project i, i=1,...,N is assigned to worker j, j=1,...,N

LP: $$\max_{x_{i,j} \in \{0,1\}^{N \times N}} \sum_{i=1,\dots,N,\ j=1,\dots,N} w_{i,j} \cdot x_{i,j}$$

s.t. $$\sum_{i=1,\dots,N} x_{i,j} = 1$$ for all j=1,...,N (each worker gets 1 project)

$$\sum_{j=1,\dots,N} x_{i,j} = 1$$ for all i=1,...,N (each project is assigned)

Next Week

Homework: Check out AMPL's student version

https://ampl.com/try-ampl/download-a-free-demo/

Review the Examples

Outlook:

- More Examples, e.g., Mixed Strategy Equilibria (Game Theory)
- Tricks to linearize Nonlinearities
- Implementations and Solvers
- Multi-objective/Penalty Approaches

Overview

Week	Dates	Topic	
1	April 27/30	Introduction + Linear Programming	
2	**May 4**/ (7)	**Linear Programming II**	
3	May 11/14	Exercise Implementations	
4	May 18	Linear + Logistic Regression	(Thu May 21 “Himmelfahrt”)
5	May 25/28	Dynamic Programming	(Mon June 1 “Pfingstmontag”)
6	June 4	Dynamic Pricing Competition	
7	June 8/11	Project Assignments	
8	June 15/18	Robust + Nonlinear Optimization	
9	June 22/25	Work on Projects: Input/Support	
10	June 29/2	Work on Projects: Input/Support	
11	July 6/9	Work on Projects: Input/Support	
12	July 13/16	Work on Projects: Input/Support	
13	July/Aug	Finish Documentation (Deadline: Aug 31)	